Non-premptive Fixed Priority schedulingwith FIFO arbitration:uniprocessor and distributed cases

Steven Martin, Pascale Minet, Laurent George

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Non-premptive Fixed Priority scheduling with FIFO arbitration: uniprocessor and distributed cases

Steven MARTIN — Pascale MINET — Laurent GEORGE

N° 5051

December 2003

THÈME 1

ISSN 0249-6399 ISRN INRIA/RR--5051--FR+ENG

Non-premptive Fixed Priority scheduling with FIFO arbitration: uniprocessor and distributed cases

Steven MARTIN* , Pascale MINET† , Laurent GEORGE‡

Thème 1 — Réseaux et systèmes
Projet HIPERCOM

Rapport de recherche n° 5051 — December 2003 — 27 pages

Abstract: In this paper, we focus on non-preemptive *Fixed Priority* scheduling. We are interested in the worst case response time of flows, both in uniprocessor and distributed cases. On a processor, the number of available priorities is generally limited. If this number is less than the number of flows to be considered, several flows have to share the same priority. Such flows are assumed to be scheduled arbitrarily in the classical approach. We assume in this paper that these flows are scheduled FIFO. This assumption leads us to revisit classical results in the uniprocessor case. As we obtain response times less than or equal to the classical results, any flow set feasible with the classical approach is feasible with our approach. The converse is false, as shown by an example. Moreover, we determine the conditions leading to shorter response times. We then establish new results in a distributed context. We show how to compute an upper bound on the end-to-end response time of any flow. For this, we use a worst case analysis based on the trajectory approach.

Key-words: Fixed priority scheduling, worst case response time, end-to-end response time, deterministic guarantee, quality of service, trajectory approach.

* Ecole Centrale d'Electronique, LACCSC, 53 rue de Grenelle, 75007 Paris, *steven.martin@ece.fr*
† INRIA Rocquencourt, 78153 Le Chesnay Cedex, *pascale.minet@inria.fr*
‡ Université Paris 12, LIIA, 120 rue Paul Armangot, 94400 Vitry, *george@univ-paris12.fr*

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex (France)
Téléphone : +33 1 39 63 55 11 —Télécopie : +33 1 39 63 53 30

Ordonnancement non-préemptif par priorités fixes avec arbitrage FIFO: cas uniprocesseur et distribué

Résumé : Dans ce rapport, nous nous intéressons à l'ordonnancement non-préemptif par *Priorités Fixes*. Nous montrons comment calculer le temps de réponse pire cas de flux en mono-processeur puis en environnement distribué. Sur un processeur, le nombre de priorités disponibles est généralement limité. Si ce nombre est plus petit que le nombre de flux considérés, plusieurs flux doivent partager la même priorité. De tels flux sont ordonnancés arbitrairement dans l'approche classique. Nous supposons dans ce rapport que ces flux sont ordonnancés FIFO. Cette hypothèse nous conduit à revisiter les résultats classiques en mono-processeur. Comme nous obtenons des temps de réponse inférieurs ou égaux aux résultats classiques, tout jeu de flux faisable avec l'approche classique est faisable avec notre approche. La réciproque est fausse, comme le montre l'exemple présenté. De plus, nous déterminons les conditions conduisant à des temps de réponse plus courts. Nous établissons ensuite de nouveaux résultats dans un environnement distribué. Nous montrons comment calculer un majorant du temps de réponse de bout-en-bout pour chaque flux. Pour cela, nous avons recours à une analyse pire cas basée sur l'approche par trajectoire.

Mots-clés : Ordonnancement par priorités fixes, temps de réponse pire cas, temps de réponse de bout-en-bout, garantie déterministe, qualité de service, approche par trajectoire.

Contents

1 Introduction **4**

2 Fixed Priority scheduling in a uniprocessor context **5**
2.1 Notations and preliminary results 5
2.2 Classical results 6
2.3 Flows of the same priority scheduled FIFO 7
2.3.1 Computation of the latest starting time 8
2.3.2 Analysis of the latest starting time 9
2.3.3 Worst case response time 11

3 Comparative evaluation in the uniprocessor case **12**
3.1 Example 12
3.2 Comparison with classical results 12

4 Response time computation in the distributed case **13**

5 Fixed priority scheduling in a distributed context: Line case **15**
5.1 Notations and preliminary results 16
5.2 Computation of the latest starting time 17
5.3 Analysis of the latest starting time 20
5.4 Worst case end-to-end response time 22
5.5 Evaluation of the delay due to packets belonging to $\overline{hp}(i)$ 23

6 Comparative evaluation in the distributed case **24**
6.1 Worst case end-to-end response time by the holistic approach 24
6.2 Examples 24

7 Conclusion **26**

1 Introduction

Fixed Priority scheduling has been extensively studied in the last years [1, 2]. It exhibits interesting properties:

- The impact of a new flow τ_i is limited to flows having priorities smaller than this of τ_i.
- It is easy to implement.
- It is well adapted for service differentiation: flows with high priorities have smaller response times.

In this paper, we focus on non-preemptive Fixed Priority scheduling. Indeed, with regard to flow scheduling, the assumption generally admitted is that packet transmission is not preemptive. We compute the worst case response time obtained with Fixed Priority scheduling. As the number of available priorities is limited on a processor, several flows have to share the same priority if this number is less than the number of flows considered. We propose, unlike the classical approach that arbitrarily schedules flows sharing the same priority, to schedule such flows according to FIFO. That is why we first revisit classical results in the uniprocessor case. Our solution enables to improve the worst case response times of such flows. As we will see, a packet cannot be delayed by other packets of the same priority released after it.

Then, we focus on the distributed case. The use of Fixed Priority scheduling in such a case presents the additional advantage of not requiring clock synchronization, unlike *Earliest Deadline First* [3, 4]. We extend our results assuming that all the flows follow the same sequence of nodes (the same *line*). Our results on the worst case end-to-end response time obtained with Fixed Priority scheduling can be used in various configurations:

- In a *Differentiated Services* architecture [5], several classes are defined, each having its own priority. The highest priority class, that is the *Expedited Forwarding* (EF) class, is scheduled Fixed Priority with the other classes. Moreover, if packets belonging to the EF class need to be differentiated, different priorities can be assigned to these packets. Hence, a Fixed Priority scheduling can be used to provide the requested differentiation.
- In an *Integrated Services* architecture [6], a priority is assigned to each flow. Fixed Priority scheduling is used to provide shorter response times to high priority flows.
- In an *hybrid* architecture, some flows are managed per class, whereas others are managed individually.

In the three configurations mentionned, several flows can share the same priority. Assuming that flows sharing the same priority are scheduled FIFO, we show how to compute a bound on the end-to-end response time of any flow, using the trajectory approach.

The rest of the paper is organized as follows. In section 2, we revisit classical results on non-preemptive Fixed Priority scheduling in the uniprocessor case, assuming that flows sharing the same priority are scheduled FIFO. In Section 3, we illustrate the benefit brought by our approach with an example and give the condition for which the worst case response time is improved. We then focus on the distributed case. Section 4 briefly discusses related work in the computation of worst case end-to-end response time. In section 5, we show how to compute an upper bound on the end-to-end response time of any flow, based on a worst case analysis, when all the flows follow the same sequence of nodes (the same *line*). Then, in Section 6, we compare our results, obtained by applying the trajectory approach, with the exact worst case end-to-end response times and with the results provided by the holistic approach. The exact values are obtained by a simulation tool we have designed. This simulation tool does an exhaustive analysis. Finally, we conclude the paper in section 7.

In this paper, we assume that time is discrete. [7] shows that results obtained with a discrete scheduling are as general as those obtained with a continuous scheduling when all flow parameters are multiples of the node clock tick. In such conditions, any set of flows is feasible with a discrete scheduling if and only if it is feasible with a continuous scheduling.

2 Fixed Priority scheduling in a uniprocessor context

We first recall in Section 2.1 some notations and preliminary results used in hard real-time scheduling. Then, we give in Section 2.2 several properties and results established in the uniprocessor case for the non-preemptive Fixed Priority scheduling algorithm. Finally, in Section 2.3, we assume that when several flows share the same priority, they are scheduled FIFO. This assumption leads us to revisit classical results in the uniprocessor case. The worst case response time of any flow is then improved.

2.1 Notations and preliminary results

We consider a set $\tau = \{\tau_1, ..., \tau_n\}$ of n sporadic flows. Each flow τ_i is defined by:

- T_i, the minimum interarrival time (called period) between two successive packets of τ_i;
- C_i, the maximum processing time of a packet of τ_i;
- J_i, the maximum release jitter of packets of τ_i.

This characterization is well adapted to real-time flows (e.g. process control, voice and video, sensor and actuator). Moreover, for any packet g, we denote $\tau(g)$ the index number of the flow which g belongs to.

Due to the scheduling model, to any flow τ_i is assigned a fixed priority P_i. Then, we denote:

- $hp(i) = \{j \in [1, n], j \neq i, \text{ such that } P_j \geq P_i\}$;
- $\overline{hp}(i) = \{j \in [1, n] \text{ such that } P_j < P_i\}$.

As said in the Introduction, packet scheduling is non-preemptive. Hence, despite the high priority of any packet m, a packet with a lower priority can delay m processing due to non-preemption. Indeed, if a packet m of any flow τ_i is released while a packet m' belonging to $\overline{hp}(i)$ is being processed, m has to wait until m' completion. It is important to notice that the non-preemptive effect is not limited to this waiting time. Indeed, the delay incurred by packet m directly due to m' may lead to consider packets with a higher priority than m released after m but before m starts its execution. The following lemma establishes the maximum delay incurred by a packet directly due to another packet with a lower priority.

Lemma 1 *In the uniprocessor case, the maximum delay incurred by any packet of any flow τ_i directly due to packets $\in \overline{hp}(i)$ is equal to:* $\max(0\,;\, C_{\overline{max},i} - 1)$, *where* $C_{\overline{max},i} = \max_{j \in \overline{hp}(i)}\{C_j\} - 1$ *if* $\overline{hp}(i) \neq \emptyset$, 0 *otherwise.*

Proof: See [4]. ■

We now recall some definitions about busy periods.

Definition 1 *An idle time t is a time such that all packets arrived before t have been processed at time t.*

Definition 2 *A busy period is defined by an interval $[t, t')$ such that t and t' are both idle times and there is no idle time $\in (t, t')$.*

Definition 3 *An idle time t of level i is a time such that all packets with a priority greater than or equal to i and arrived before t have been processed at time t.*

Definition 4 *A busy period of level i is defined by an interval $[t, t')$ such that t and t' are both idle times of level i and there is no idle time of level $i \in (t, t')$.*

Let $W_i(t)$ denote the latest starting time of the instance of τ_i released at time t.
Let R_i denote the worst case response time of flow τ_i.

2.2 Classical results

Classical results have been established assuming that flows sharing the same priority are scheduled arbitrarily. We recall them here, in order to make a comparison with our results, established when flows sharing the same priority are scheduled FIFO. A packet m of flow τ_i, generated at time t, can be delayed by:

- packets of flows in $hp(i)$ arrived at a time less than or equal to $W_i(t)$;
- a packet of a flow in $\overline{hp}(i)$.

Property 1 *In the uniprocessor case, when all the flows are scheduled according to the fixed priority algorithm, the worst case response time of any flow τ_i meets:*

$R_i = \max_{k=0..K} \{W_i(k \cdot T_i - J_i) - k \cdot T_i\} + C_i + J_i$, *where:*

$W_i(k \cdot T_i - J_i) = \sum_{j \in hp(i)} \left(1 + \left\lfloor \frac{W_i(k \cdot T_i - J_i) + J_j}{T_j} \right\rfloor\right) \cdot C_j + k \cdot C_i + \max(0\,;\, C_{\overline{max},i} - 1)$,

with K the smallest integer value such that $W_i(K \cdot T_i - J_i) + C_i + J_i \leq (K+1) \cdot T_i$.

Proof: See [4]. ■

2.3 Flows of the same priority scheduled FIFO

Unlike the classical approach, we now adopt Assumption 1.

Assumption 1 *Flows sharing the same priority are scheduled* FIFO.

With this assumption, the worst case response time of any flow can be improved. A packet m of flow τ_i, generated at time t, can no longer be delayed by packets of other flows having the same priority as τ_i arrived after m. Hence, m processing is delayed by:

- packets of flows having a priority strictly greater than P_i and arrived at a time $\leq W_i(t)$;
- packets of flows having a priority equal to P_i and arrived at a time $\leq t$;
- a packet of a flow in $\overline{hp}(i)$;

Then, we denote:

- $gp(i) = \{j \in [1,n]$, such that $P_j > P_i\}$;
- $sp(i) = \{j \in [1,n], j \neq i$, such that $P_j = P_i\}$;
- $\overline{hp}(i) = \{j \in [1,n]$ such that $P_j < P_i\}$.

Definition 5 *For any flow τ_i, the processor utilization factor for the flows belonging to $gp(i)$ is denoted $U_{gp(i)}$. It is the fraction of processor time spent to process packets belonging to $gp(i)$. It is equal to $\sum_{j \in gp(i)} (C_j/T_j)$.*

We determine in Subsection 2.3.1 the latest starting time of any packet m belonging to flow τ_i. This time can be computed as a limit of a series, that we study in Subsection 2.3.2. Thanks to this analysis, we give in Subsection 2.3.3 an upper bound on the worst case response time of flow τ_i.

2.3.1 Computation of the latest starting time

We first determine the scenarios that lead to the worst case response time of flow τ_i.

Lemma 2 *In the uniprocessor case, when all the flows are scheduled according to the Fixed Priority algorithm, flows having the same priority being scheduled FIFO, then the worst case response time of any flow τ_i is reached in the first busy period of a scenario where:*

- *$\forall$ flow $\tau_j \in gp(i) \cup sp(i)$, the first packet of τ_j is released at time $-J_j$ and all other packets of flow τ_j are periodic, with period T_j;*
- *a packet of the flow τ_j belonging to $\overline{hp}(i)$, if any, with the maximum processing time is released at time -1;*
- *the first packet of flow τ_i is released at time t_i^0, with $-J_i \leq t_i^0 < -J_i + T_i$ and all other packets of τ_i are periodic, with period T_i.*

Proof: Let us consider the first busy period of a scenario I in which a packet m of flow τ_i, released at time t, is processed. The processing of packet m can be delayed by:

1. packets belonging to flow $\tau_j \in gp(i)$, released before m starts its execution.
2. packets belonging to flow $\tau_j \in sp(i)$, released before time t.
3. packets belonging to flow τ_i, released before m.
4. a packet with a priority lower than P_i.

We now modify scenario I to worsen the response time of packet m. The workload generated by packets of the first and second category is maximized when all flows $\tau_j \in gp(i) \cup sp(i)$ are first released at time $-J_j$ and then periodically, with period T_j. In the same way, the workload generated by packets of the third category is maximized when the first packet of flow τ_i is released at time t_i^0, with $-J_i \leq t_i^0 < -J_i + T_i$ and all other packets of τ_i are periodic, with period T_i. Finally, to maximize the delay due to a packet belonging to $\overline{hp}(i)$, a packet of the flow $\tau_j \in \overline{hp}(i)$, if any, with the maximum processing time is released at time -1. In this new scenario, the response time of packet m is either left unchanged or worst. ■

Lemma 3 *In the uniprocessor case, when all the flows are scheduled according to the Fixed Priority algorithm, flows having the same priority being scheduled FIFO, then for any packet of any flow τ_i, released at time t, its latest starting time is given by:*

$$W_i(t) = \sum_{j \in gp(i)} \left(1 + \left\lfloor \frac{W_m(t) + J_j^1}{T_j} \right\rfloor\right) \cdot C_j + \sum_{j \in sp(i)} \left(1 + \left\lfloor \frac{t + J_j^1}{T_j} \right\rfloor\right) \cdot C_j + \left\lfloor \frac{t + J_i^1}{T_i} \right\rfloor \cdot C_i + \max(0\,;\, C_{\overline{max},i} - 1).$$

Proof: To compute the latest starting time of packet m, we identify the busy period of level P_i that affect the delay of m, that is the one in which m is processed. We define f as the first packet processed in this busy period with a priority greater than or equal to P_i. Due to the non-preemptive effect, the execution of f can be delayed once by a packet with a priority less than P_i.

For the sake of simplicity, we number consecutively the packets of the considered busy period of level P_i. Hence, we denote $m' - 1$ (resp. $m' + 1$) the packet preceding (resp. succeeding) m'. Moreover, we consider the release time of packet f as the time origin. The latest starting time of packet m is then equal to the processing time of packets f to $m - 1$, plus $\max(0\,;\, C_{\overline{max},i} - 1)$. Then, we get: $W_i(t) = \sum_{g=f}^{m} C_{\tau(g)} - C_i + \max(0\,;\, C_{\overline{max},i} - 1)$.

The term $\sum_{g=f}^{m} C_{\tau(g)}$ is bounded by the maximum workload generated by flows belonging to $gp(i)$ in the interval $[0, W_i(t)]$, plus the maximum workload generated by flows belonging to $sp(i)$ in the interval $[0, t]$, plus the maximum workload generated by packets of flow τ_i in the interval $[0, t]$. By definition, the maximum workload generated by:

- flows $\tau_j \in gp(i)$ in the interval $[0, W_i(t)]$ is equal to: $\sum_{j \in gp(i)} (1 + \lfloor (W_i(t) + J_j)/T_j \rfloor) \cdot C_j$.
- flows $\tau_j \in sp(i)$ in the interval $[0, t]$ is equal to: $\sum_{j \in sp(i)} (1 + \lfloor (t + J_j)/T_j \rfloor) \cdot C_j$.
- flow τ_i in the interval $[0, t]$ is equal to: $1 + \lfloor (t + J_i)/T_i \rfloor) \cdot C_i$. ■

2.3.2 Analysis of the latest starting time

We now focus on the following series that we denote $\mathcal{W}_i$:

$$\begin{cases} W_i^{(0)}(t) = \sum_{j \in gp(i)} C_j + \sum_{j \in sp(i)} \left(1 + \left\lfloor \frac{t + J_j}{T_j} \right\rfloor\right) \cdot C_j + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \cdot C_i + \max(0\,;\, C_{\overline{max},i} - 1) \\ W_i^{(p+1)}(t) = \sum_{j \in gp(i)} \left(1 + \left\lfloor \frac{W_i^{(p)}(t) + J_j}{T_j} \right\rfloor\right) \cdot C_j + \sum_{j \in sp(i)} \left(1 + \left\lfloor \frac{t + J_j}{T_j} \right\rfloor\right) \cdot C_j + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \cdot C_i + \max(0; C_{\overline{max},i} - 1). \end{cases}$$

To prove the existence of $W_i(t)$, solution of the equation given in Lemma 3, we show that if Condition 1 is met, $\mathcal{W}_i$ is convergent. Then, Lemmas 4 and 5 show that only a limited set of release times of flow τ_i has to be tested to obtain the latest starting time of a flow packet.

Condition 1 *If $U_{gp(i)} < 1$, where $U_{gp(i)}$ denotes the utilization factor for the flows belonging to $gp(i)$, then the series $\mathcal{W}_i$ is convergent.*

Proof: The series $\mathcal{W}_i$ is a non-decreasing series as the floor function is non-decreasing. Moreover, this series is upper bounded by: $X/(1-U_{gp(i)})$, where:

$$X = \sum_{j\in gp(i)} \left(1+\frac{J_j}{T_j}\right)\cdot C_j + \sum_{j\in sp(i)} \left(1+\left\lfloor\frac{t+J_j}{T_j}\right\rfloor\right)\cdot C_j + \left\lfloor\frac{t+J_i}{T_i}\right\rfloor\cdot C_i + \max(0\,;\, C_{\overline{max},i}-1).$$

Indeed, by recurrence, we have: $W_i^{(0)}(t) \leq X$, that is less than or equal to: $X/(1-U_{gp(i)})$, assuming $U_{gp(i)} < 1$. We now assume that the recurrence is true at rank p and show that it is true at rank $p+1$. Then, $W_i^{(p+1)}(t)$ meets:

$$\sum_{j\in gp(i)} \left(1+\left\lfloor\frac{W_i^{(p)}(t)+J_j}{T_j}\right\rfloor\right)\cdot C_j + \sum_{j\in sp(i)} \left(1+\left\lfloor\frac{t+J_j}{T_j}\right\rfloor\right)\cdot C_j + \left\lfloor\frac{t+J_i}{T_i}\right\rfloor\cdot C_i + \max(0\,;\, C_{\overline{max},i}-1),$$

$$\leq\ W_i^{(p)}(t)\cdot \sum_{j\in gp(i)} \frac{C_j}{T_j} + \sum_{j\in gp(i)} \left(1+\frac{J_j}{T_j}\right)\cdot C_j + \sum_{j\in sp(i)} \left(1+\left\lfloor\frac{t+J_j}{T_j}\right\rfloor\right)\cdot C_j + \left\lfloor\frac{t+J_i}{T_i}\right\rfloor\cdot C_i + \max(0; C_{\overline{max},i}-1),$$

$$\leq\ W_i^{(p)}(t)\cdot U_{gp(i)} + X \quad\leq\quad X\cdot U_{gp(i)}/(1-U_{gp(i)}) + X \quad=\quad X/(1-U_{gp(i)}).$$

The series $\mathcal{W}_i$ is non-decreasing and upper bounded. Hence, this series is convergent. ■

Lemma 4 *In the uniprocessor case, when all flows are scheduled according to the fixed priority algorithm, flows having the same priority being scheduled FIFO, the worst case response time of any flow τ_i is obtained for a packet released at time $t \in \mathcal{S}$, where $\mathcal{S}$ is the ordered set of times $t = k_j\cdot T_j - J_j \geq -J_i$, $j \in sp(i)\cup\{i\}$ and $k_j \in \mathbb{N}$.*

Proof: We consider the series $\mathcal{W}_i$ and prove the lemma by recurrence. By assumption, no packet of τ_i can be released before time $-J_i$. Hence, we only consider times $t \geq -J_i$. By definition, if t_1 and t_2 are two consecutive times of set $\mathcal{S}$: $\forall\, j \in sp(i)\cup\{i\}$, $\forall\, t' \in [t_1, t_2)$, $\lfloor (t'+J_j)/T_j\rfloor = \lfloor (t_1+J_j)/T_j\rfloor$. Hence, the lemma is met at rank 0: $W_i^{(0)}(t') = W_i^{(0)}(t_1)$. Assuming that the recurrence is true at rank p, we show that it is true at rank $p+1$. Indeed, $W_i^{(p+1)}(t')$ equals:

$$\sum_{j\in gp(i)}\left(1+\left\lfloor\frac{W_i^{(p)}(t')+J_j}{T_j}\right\rfloor\right)\cdot C_j+\sum_{j\in sp(i)}\left(1+\left\lfloor\frac{t'+J_j}{T_j}\right\rfloor\right)\cdot C_j+\left\lfloor\frac{t'+J_i}{T_i}\right\rfloor\cdot C_i+\max(0\,;\,C_{\overline{max},i}-1)$$

$$=\sum_{j\in gp(i)}\left(1+\left\lfloor\frac{W_i^{(p)}(t_1)+J_j}{T_j}\right\rfloor\right)\cdot C_j+\sum_{j\in sp(i)}\left(1+\left\lfloor\frac{t_1+J_j}{T_j}\right\rfloor\right)\cdot C_j+\left\lfloor\frac{t_1+J_i}{T_i}\right\rfloor\cdot C_i+\max(0\,;\,C_{\overline{max},i}-1)$$

$$=W_i^{(p+1)}(t_1).$$ ■

Lemma 5 *In the uniprocessor case, when all flows are scheduled according to the fixed priority algorithm, flows having the same priority being scheduled FIFO, if a packet of τ_i released at time $t\in\mathcal{S}$ meets: $W_i(t)+C_i\leq t+T_i$, then it is useless to compute the response time of a packet released at time $t+\alpha\cdot T_i$, $\alpha\in\mathbb{N}^*$.*

Proof: Let us consider a fixed $j\in sp(i)\cup\{i\}$. Let m' be the packet of flow τ_i released at time $t\in\mathcal{S}$. We compute the response time of m' and distinguish two cases:

- If $W_i(t)+C_i>t+T_i$, the packet of τ_i released at time $t+T_i$ is delayed by the processing of previous packets of τ_i. Hence, the response time of this packet must be computed.
- If $W_i(t)+C_i\leq t+T_i$, the packet of τ_i released at time $t+T_i$ is processed without being delayed by previous packets of τ_i. According to Lemma 2, this packet is not processed in the worst case conditions. Hence, the worst case response time of τ_i will not be reached by this packet. ■

2.3.3 Worst case response time

With the previous lemmas, we can now establish the following property, that gives the worst case response time of any flow τ_i.

Property 2 *In the uniprocessor case, when all flows are scheduled according to the fixed priority algorithm, flows having the same priority being scheduled FIFO, and condition 1 is met, the worst case response time of any flow τ_i is equal to:*

$R_i=\max_{t\in\mathcal{S}'}\{W_i(t)-t\}+C_i$, *where:*

$$W_i(t)=\sum_{j\in gp(i)}\left(1+\left\lfloor\frac{W_i(t)+J_j}{T_j}\right\rfloor\right)\cdot C_j+\sum_{j\in sp(i)}\left(1+\left\lfloor\frac{t+J_j}{T_j}\right\rfloor\right)\cdot C_j+\left\lfloor\frac{t+J_i}{T_i}\right\rfloor\cdot C_i+\max(0\,;\,C_{\overline{max},i}-1),$$

$\mathcal{S}'$ denotes the ordered set of times $t=k_j\cdot T_j-J_j\geq -J_i$, excluding the times $t'=k_j\cdot T_j-J_j+\alpha\cdot T_i$ such that $W_i(t')+C_i\leq t'+T_i$, with $j\in sp(i)\cup\{i\}$, $k_j\in\mathbb{N}$ and $\alpha\in\mathbb{N}^*$.

Proof: This property can be deduced from Lemmas 3, 4 and 5. ■

3 Comparative evaluation in the uniprocessor case

In this section, we present an example that illustrates the improvement we get on the worst case response time when applying Assumption 1. Then, Property 3 highlights the benefit brought by a FIFO scheduling of flows sharing the same priority.

3.1 Example

In this example, we consider five flows, without release jitter, whose characteristics are given in Table 1. The load is equal to 100%. There is only one flow in the priorities 3 and 2, whereas three flows share the lowest priority 1.

Table 1. Improvement on the worst case response time

					Our results		*Classical results*	
Flow	P_i	C_i	T_i	D_i	$W_i(t)$	R_i	$W_i(t)$	R_i
τ_1	1	4	20	30	24	28	32	36
τ_2	1	4	20	30	24	28	32	36
τ_3	1	4	20	30	24	28	32	36
τ_4	2	4	20	15	11	15	11	15
τ_5	3	8	40	11	3	11	3	11

As expected, results show that for priorities 2 and 3, there is no difference between our results and the classical ones. Indeed, there is only one flow for each of these two priorities. Concerning the lowest priority, we get a worst case response time of 28, whereas the classical response time is 36. We get an improvement higher than 22%.

As with our approach, all flows meet their deadline, this flow set is feasible. However, it is not feasible with the classical approach as the flows τ_1, τ_2 and τ_3 do not meet their deadline 30. This example shows the interest of our approach.

3.2 Comparison with classical results

Property 3 highlights the benefit brought by a FIFO scheduling of flows sharing the same priority by establishing the conditions that lead to shorter response times. Table 2 summarizes the worst case response time of any flow τ_i when flows sharing the same priority are scheduled (i) arbitrarily, denoted classical results and (ii) FIFO, denoted our results.

Property 3 *The response time of any flow τ_i is shorter than the classical result as soon as:*

$$\exists\, k \in [0, K],\ W_i(k \cdot T_i - J_i) \geq \max\left(\min_{j \in sp(i)}(T_j)\,; k \cdot T_i - J_i\right),$$

where K denotes the smallest integer value such that: $W_i(K \cdot T_i - J_i) + C_i \leq (K+1) \cdot T_i - J_i$.

Proof: Let us consider packet m of flow τ_i released at time $t = k \cdot T_i - J_i$, with $k \in [0, K]$. If $W_i(t) \geq \max\left(\min_{j \in sp(i)}(T_j)\,;t\right)$, then, with the classical approach, m is delayed by packets of flow $\tau_j \in sp(i)$, released after time t and before $W_i(t)$. With FIFO scheduling for flows of the same priority, m would not have been delayed by these packets released after it. ■

Table 2. Response time in the uniprocessor case with and without Assumption 1

Classical results
$R_i = \max_{k=0..K}\{W_i(k \cdot T_i - J_i) - k \cdot T_i\} + C_i + J_i$, where:
$W_i(k \cdot T_i - J_i) = \sum_{j \in hp(i)} \left(1 + \left\lfloor \frac{W_i(k \cdot T_i - J_i) + J_j}{T_j} \right\rfloor\right) \cdot C_j + k \cdot C_i + \max\left(0\,;\, C_{\overline{max},i} - 1\right)$,
with K the smallest integer value such that $W_i(K \cdot T_i - J_i) + C_i + J_i \leq (K+1) \cdot T_i$.
Our results
$R_i = \max_{t \in \mathcal{S}'}\{W_i(t) - t\} + C_i$, where:
$W_i(t) = \sum_{j \in gp(i)} \left(1 + \left\lfloor \frac{W_i(t) + J_j}{T_j} \right\rfloor\right) \cdot C_j + \sum_{j \in sp(i)} \left(1 + \left\lfloor \frac{t + J_j}{T_j} \right\rfloor\right) \cdot C_j + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \cdot C_i + \max(0\,;\, C_{\overline{max},i} - 1)$,
with $\mathcal{S}'$ the ordered set of times $t = k_j \cdot T_j - J_j \geq -J_i$, excluding the times $t' = k_j \cdot T_j - J_j + \alpha \cdot T_i$ such that $W_i(t') + C_i \leq t' + T_i$, $j \in sp(i) \cup \{i\}$, $k_j \in \mathbb{N}$ and $\alpha \in \mathbb{N}^*$.

As with our approach, we obtain response times less than or equal to the classical results, any flow set feasible with the classical approach is feasible with our approach. The converse is false, as shown by the example given in Section 3.1.

4 Response time computation in the distributed case

To determine the maximum end-to-end response time, several approaches can be used: a stochastic or a deterministic one. A *stochastic approach* consists in determining the mean behavior of the considered network, leading to mean, statistical or probabilistic end-to-end response times [8, 9]. A *deterministic approach* is based on a worst case analysis of the network behavior, leading to worst case end-to-end response times [10, 11].

In this paper, we are interested in the deterministic approach as we want to provide a deterministic guarantee of worst case end-to-end response times for any flow in the network. In this context, two different approaches can be used to determine the worst case end-to-end delay: the holistic approach and the trajectory approach.

• The *holistic approach* [12] considers the worst case scenario on each node visited by a flow, accounting for the maximum possible jitter introduced by the previous visited nodes. If no jitter control is done, the maximum jitter will increase throughout the visited nodes. In this case, the minimum and maximum response times on a node h induce a maximum jitter on the next visited node $h+1$ that leads to a worst case response time and then a maximum jitter on the following node and so on. Otherwise, the jitter can be either cancelled or constrained.

- the *Jitter Cancellation technique* consists in cancelling, on each node, the jitter of a flow before it is considered by the node scheduler [11]: a flow packet is held until its latest possible reception time. Hence a flow packet arrives at node $h+1$ with a jitter depending only on the jitter introduced by the previous node h and the link between them. As soon as this jitter is cancelled, this packet is seen by the scheduler of node $h+1$. The worst case end-to-end response time is obtained by adding the worst case response time, without jitter (as cancelled) on every node;
- the *Constrained Jitter technique* consists in checking that the jitter of a flow remains bounded by a maximum acceptable value before the flow is considered by the node scheduler. If not, the jitter is reduced to the maximum acceptable value by means of traffic shaping.

As a conclusion, the holistic approach can be pessimistic as it considers worst case scenarios on every node possibly leading to impossible scenarios.

• The *trajectory approach* [13] consists in examining the scheduling produced by all the visited nodes of a flow. In this approach, only possible scenarios are examined. For instance, the fluid model (see [14] for GPS) is relevant to the trajectory approach. This approach produces the best results as no impossible scenario is considered but is somewhat more complex to use. This approach can also be used in conjunction with a jitter control (see [15] for EDF, and [14] for GPS). In this paper, we adopt the trajectory approach without jitter control in a distributed system to determine the maximum end-to-end response time of a flow.

We can also distinguish two main traffic models: the sporadic model and the token bucket model. The sporadic model has been used in the holistic approach and in the trajectory approach, while the token bucket model has been used only in the trajectory approach.

• The *sporadic model* is classically defined by three parameters: the maximum processing time, the minimum interarrival time and the maximum release jitter, (see section 5). This model is natural and well adapted for real-time applications.

• The *token bucket* [10, 14, 15] is defined by two parameters: σ, the bucket size and ρ, the token throughput. The token bucket can model a flow or a flow aggregate. In the first case, it requires to maintain per flow information on every visited node. This solution is not scalable. In the second case, the choice of good values for the token bucket parameters is complex

when flows have different characteristics. A bad choice can lead to bad response times, as the end-to-end response times strongly depend on the choice of the token bucket parameters [15, 16]. Furthermore, the token bucket parameters can be optimized for a given configuration, only valid at a given time. If the configuration evolves, the parameters of the token bucket should be recomputed on every node to remain optimal. This is not generally done.

In this paper, we adopt the trajectory approach with the sporadic traffic model and we establish new results that we compare with those provided by the classical holistic approach.

5 Fixed priority scheduling in a distributed context: Line case

We investigate the problem of providing a deterministic end-to-end response time guarantee to any flow in a distributed system. The end-to-end response time of a flow is defined between its ingress node and its egress node. We want to provide an upper bound on the end-to-end response time of any flow. As we make no particular assumption concerning the arrival times of packets in the distributed system, the feasibility of a set of flows is equivalent to meet the requirement, whatever the arrival times of the packets in the distributed system. Moreover, we assume the following models.

Scheduling model

We consider that all the nodes in the distributed system schedule packets according to the fixed priority algorithm. Moreover, we assume that packet scheduling is non-preemptive. Therefore, the node scheduler waits for the completion of the current packet transmission (if any) before selecting the next packet.

Network model

We consider a distributed system where links interconnecting nodes are supposed to be FIFO and the network delay between two nodes has known lower and upper bounds: L_{min} and L_{max}. Moreover, we consider neither network failures nor packet losses.

Traffic model

We consider a set $\tau = \{\tau_1, ..., \tau_n\}$ of n sporadic flows. Each flow τ_i follows a sequence of nodes whose first node is the ingress node of the flow. In the following, we call *line* this sequence. Moreover, a sporadic flow τ_i following a line $\mathcal{L}$ consisted of q nodes, numbered from 1 to q, is defined by:

- T_i, the minimum interarrival time (called period) between two successive packets of τ_i;
- C_i^h, the maximum processing time on node h of a packet of τ_i;
- J_i^1, the maximum jitter of packets of τ_i arriving in the distributed system.

We consider that all flows follow the same line $\mathcal{L}$ in the distributed system, that is the same sequence of nodes consisting of q nodes numbered from 1 to q. To determine the end-to-end response time of any packet m belonging to any flow τ_i, we first extend the worst case analysis performed in Section 2.3 in order to compute the latest starting time of paket m in node q, the last node visited. Then, the mathematical expression of this latest starting time, that is an iterative expression, is analyzed. Finally, we propose a bound on the end-to-end response time.

5.1 Notations and preliminary results

In addition to the notations presented in Section 2, we use the following ones:

$\mathcal{L}$	line consisting of q nodes numbered from 1 to q and followed by all the flows;
slow	the slowest node of line $\mathcal{L}$, that is: $\forall$ flow τ_j, $\forall$ node $h \in \mathcal{L}$, $C_j^h \leq C_j^{slow}$;
$W_i^q(t)$	the latest starting time of the packet of τ_i entered the system at time t;
$R_i^{\mathcal{L}}$	the worst case end-to-end response time of flow τ_i in the distributed system;
$M_i^{1,q}$	the minimum time taken by a packet of flow τ_i to go from node 1 to node q;
$L_m^{h,h+1}$	the network delay experienced by packet m between nodes h and $h+1$;
$H_i^{1,h}$	the maximum delay incurred by flow τ_i directly due to flows $\in \overline{hp}(i)$ while visiting nodes 1 to h.

Moreover, on any node h, the processing time of any packet m' is less than or equal to:

- $C^h_{max,i} = \max_{j/P_j \geq P_i}\{C_j^h\}$ if the priority of m' is greater than or equal to P_i;
- $C^h_{\overline{max},i} = \max_{j/P_j < P_i}\{C_j^h\}$ if the priority of m' is less than P_i.
  If there is no flow belonging to $\overline{hp}(i)$, then $C^h_{\overline{max},i} = 0$.

We now recall the definition of the processor utilization factor for a set of flows.

Definition 6 *For any node h, for any flow τ_i visiting h, the processor utilization factor for the flows belonging to $gp(i)$ is denoted $U^h_{gp(i)}$. It is the fraction of processor time spent by node h to process packets belonging to $gp(i)$. It is equal to $\sum_{j \in gp(i)}(C_j^h/T_j)$.*

As explained in Section 2.1, any packet m of flow τ_i can be delayed by a packet with a lower priority due to non-preemption. The following lemma gives an upper bound on the maximum delay incurred by m and directly due to packets belonging to $\overline{hp}(i)$.

Lemma 6 *When all the flows follow the same line $\mathcal{L}$ consisting of q nodes numbered from 1 to q, then the maximum delay incurred by any packet m of any flow τ_i directly due to packets belonging to $\overline{hp}(i)$ meets: $H_i^{1,q} \leq \sum_{h=1}^{q} \max\left(0\,;\, C^h_{\overline{max},i} - 1\right)$, where $C^h_{\overline{max},i}$ denotes the maximum processing time on node h of a packet with a priority less than P_i.*

Proof: By definition, no packet of flows belonging to $\overline{hp}(i)$ can be processed in a busy period of level P_i, except the first packet of this busy period (cf. Lemma 1). Hence, the maximum delay incurred by any packet m of flow τ_i directly due to packets belonging to $\overline{hp}(i)$ can not be greater than $\max\left(0\,;\, C^h_{\overline{max},i} - 1\right)$ on each node h visited. ■

5.2 Computation of the latest starting time

Lemma 7 *When all the flows follow the same line $\mathcal{L}$ consisting of q nodes numbered from 1 to q and these flows are scheduled according to the Fixed Priority algorithm, flows having the same priority being scheduled FIFO, then for any packet belonging to any flow τ_i, entered the distributed system at time t, its latest starting time in node q is given by:*

$$W_i^q(t) = \sum_{j \in hp(i)} \left(1 + \left\lfloor \frac{\max\left(0\,;\, W_i^q(t) - M_j^{1,q}\right) + J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + \sum_{j \in sp(i) \cup \{i\}} \left(1 + \left\lfloor \frac{t + J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + \mathcal{A}_i,$$

where $\mathcal{A}_i = \sum^{q}_{\substack{h=1 \\ h \neq slow}} C^h_{max,i} - C_i^q + H_i^{1,q} + (q-1) \cdot Lmax.$

Proof: To determine the latest starting time of packet m, we identify the busy periods of level P_i that affect the delay of m. For this, we consider the busy period of level P_i, denoted bp_i^q, in which m is processed on node q and we define $f(q)$ as the first packet processed in bp_i^q with a priority greater than or equal to P_i. Due to the non-preemptive effect, the execution of $f(q)$ can be delayed once by a packet with a priority less than P_i.

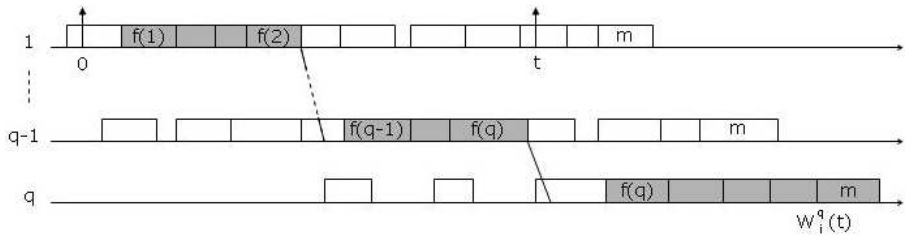

Figure 1: Starting time of packet m in node q

The packet $f(q)$ has been processed in a busy period on node $q-1$ at least of level P_i. Let bp_i^{q-1} be this busy period. We then define $f(q-1)$ as the first packet processed in bp_i^{q-1} with a priority greater than or equal to P_i. And so on until the busy period of node 1 in which the packet $f(1)$ is processed (see figure 1).
For the sake of simplicity, we number consecutively the packets of the considered busy periods of level P_i. Hence, we denote $m'-1$ (respectively $m'+1$) the packet preceding (respectively succeeding) m'. Moreover, in the following, we consider the arrival time of packet $f(1)$ in node 1 as the time origin.

By adding parts of the considered busy periods, we can now express the starting time of packet m in node q, that is:

the processing time on node 1 of packets $f(1)$ to $f(2)$ + $L^{1,2}_{f(2)}$

\+ the processing time on node 2 of packets $f(2)$ to $f(3)$ + $L^{2,3}_{f(3)}$

\+ ...

\+ the processing time on node q of packets $f(q)$ to $(m-1)$.

\+ $H_i^{1,q}$, the maximum delay directly due to packets $\in \overline{hp}(i)$ while visiting nodes 1 to q.

By convention, $f(q+1)=m$. Then, the starting time, in node q, of packet m entered the distributed system at time t meets: $W_i(t) \leq \sum_{h=1}^{q}\left(\sum_{g=f(h)}^{f(h+1)} C^h_{\tau(g)}\right) - C_i^q + H_i^{1,q} + (q-1)\cdot L_{max}$.

By definition, on any node h, packets $f(h)$ to $f(h+1)$ have a priority higher than or equal to P_i since all the busy periods we consider are at least of level P_i. We now consider the term $\sum_{h=1}^{q}(\sum_{g=f(h)}^{f(h+1)} C^h_{\tau(g)}) - C_i^q$ and distinguish the nodes visited by the flows before node *slow* and those visited after. Thus, we get $\sum_{h=1}^{q}(\sum_{g=f(h)}^{f(h+1)} C^h_{\tau(g)})$ less than or equal to:

$$\underbrace{\sum_{h=1}^{slow-1}\left(\sum_{g=f(h)}^{f(h+1)-1} C^h_{\tau(g)} + C^h_{\tau(f(h+1))}\right)}_{\textit{nodes visited before slow}} + \underbrace{\sum_{g=f(slow)}^{f(slow+1)} C^{slow}_{\tau(g)}}_{\textit{node slow}} + \underbrace{\sum_{h=slow+1}^{q}\left(\sum_{g=f(h)+1}^{f(h+1)} C^h_{\tau(g)} + C^h_{\tau(f(h))}\right)}_{\textit{nodes visited after slow}}.$$

For any node $h \in [1,q]$, for any packet m' visiting h, the processing time of m' on node h is less than $C^{slow}_{\tau(m')}$. Then, as packets are numbered consecutively from $f(1)$ to $f(q+1)=m$, we get inequation (1). By considering that on any node h, the processing time of a packet with a priority greater than or equal to P_i is less than or equal to $C^h_{max,i} = \max_{j/P_j \geq P_i}\{C^h_j\}$, we get inequation (2).

$$\sum_{h=1}^{slow-1}\left(\sum_{g=f(h)}^{f(h+1)-1} C^h_{\tau(g)}\right) + \sum_{g=f(slow)}^{f(slow+1)} C^{slow}_{\tau(g)} + \sum_{h=slow+1}^{q}\left(\sum_{g=f(h)+1}^{f(h+1)} C^h_{\tau(g)}\right) - C^q_i \leq \sum_{g=f(1)}^{m} C^{slow}_{\tau(g)} - C^q_i \quad (1)$$

$$\sum_{h=1}^{slow-1} C^h_{\tau(f(h+1))} + \sum_{h=slow+1}^{q} C^h_{\tau(f(h))} \leq \sum_{\substack{h=1 \\ h\neq slow}}^{q} C^h_{max,i} \quad (2)$$

By (1) and (2), we get: $\sum_{h=1}^{q}\left(\sum_{g=f(h)}^{f(h+1)} C^h_{\tau(g)}\right) - C^q_i \leq \sum_{g=f(1)}^{m} C^{slow}_{\tau(g)} - C^q_i + \sum_{\substack{h=1 \\ h\neq slow}}^{q} C^h_{max,i}$.

The term $\sum_{g=f(1)}^{m} C^{slow}_{\tau(g)}$ is bounded by the maximum workload generated by flows τ_j belonging to $gp(i)$ in the interval $[0, \max(0\,;\, W^q_i(t) - M^{1,q}_j)]$, plus the maximum workload generated by flows belonging to $sp(i)$ in the interval $[0,t]$, plus the maximum workload generated by flow τ_i in the interval $[0,t]$. Indeed, any packet entered the distributed system before $f(1)$ does not interfere with the considered packets processed in the selected busy periods.

It is the same for any packet belonging to $gp(i)$ and arriving in node q after m starts its execution. Then, a packet of any flow $\tau_j \in gp(i)$ does not delay m if it arrives in node 1 after the time: $W^q_i(t) - M^{1,q}_j$, where $M^{1,q}_j$ denotes the minimum time taken by a packet of flow τ_j to go from node 1 to node q. In the same way, because links are FIFO and packets of the same priority are scheduled FIFO, any packet belonging to $sp(i)$ and arriving in node 1 after m does not delay m. Thus, we get $\sum_{g=f(1)}^{m} C^{slow}_{\tau(g)}$ less than or equal to:

$$\sum_{j\in hp(i)}\left(1+\left\lfloor \frac{\max\left(0\,;W^q_i(t)-M^{1,q}_j\right)+J^1_j}{T_j}\right\rfloor\right)\cdot C^{slow}_j + \sum_{j\in sp(i)}\left(1+\left\lfloor\frac{t+J^1_j}{T_j}\right\rfloor\right)\cdot C^{slow}_j + \left(1+\left\lfloor\frac{t+J^1_i}{T_i}\right\rfloor\right)\cdot C^{slow}_i.$$

Hence, the latest starting time of packet m in node q meets:

$$W^q_i(t) = \sum_{j\in hp(i)}\left(1+\left\lfloor \frac{\max\left(0\,;W^q_i(t)-M^{1,q}_j\right)+J^1_j}{T_j}\right\rfloor\right)\cdot C^{slow}_j + \sum_{j\in sp(i)\cup\{i\}}\left(1+\left\lfloor\frac{t+J^1_j}{T_j}\right\rfloor\right)\cdot C^{slow}_j + \mathcal{A}_i,$$

where $\mathcal{A}_i = \sum_{\substack{h=1 \\ h\neq slow}}^{q} C^h_{max,i} - C^q_i + H^{1,q}_i + (q-1)\cdot Lmax$. ■

5.3 Analysis of the latest starting time

We now focus on the following series that we denote $\mathcal{W}_i^q$:

$$\begin{cases} W_i^{q^{(0)}}(t) = \sum\limits_{j \in gp(i)} C_j^{slow} + \sum\limits_{j \in sp(i) \cup \{i\}} \left(1 + \left\lfloor \frac{t+J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + \mathcal{A}_i \\ \\ W_i^{q^{(p+1)}}(t) = \sum\limits_{j \in gp(i)} \left(1 + \left\lfloor \frac{\max\left(0; W_i^{q^{(p)}}(t) - M_j^{1,q}\right) + J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + \sum\limits_{j \in sp(i) \cup \{i\}} \left(1 + \left\lfloor \frac{t+J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + \mathcal{A}_i, \end{cases}$$

with $\mathcal{A}_i = \sum_{\substack{h=1 \\ h \neq slow}}^{q} C_{max,i}^h - C_i^q + H_i^{1,q} + (q-1) \cdot L_{max}$.

As in Section 2.3.2, we first prove the existence of $W_i^q(t)$, solution of the equation given in Lemma 7. Then, we show that only a limited set of arrival times in the distributed system has to be tested to obtain the latest starting time of a flow packet in node q. We finally show how to compute the worst case end-to-end response time of any flow τ_i.

Condition 2 *If $U_{gp(i)}^{slow} < 1$, where $U_{gp(i)}^{slow}$ denotes the utilization factor on node slow for the flows belonging to $gp(i)$, then $\mathcal{W}_i^q$ is convergent.*

Proof: The series $\mathcal{W}_i^q$ is a non-decreasing series as the floor function is non-decreasing. Moreover, this series is upper bounded by: $X/(1 - U_{gp(i)}^{slow})$, where:

$$X = \sum_{j \in gp(i)} \left(1 + \frac{J_j^1}{T_j}\right) \cdot C_j^{slow} + \sum_{j \in sp(i) \cup \{i\}} \left(1 + \left\lfloor \frac{t+J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + \mathcal{A}_i.$$

Indeed, by recurrence, we have: $W_i^{q^{(0)}}(t) \leq X$, that is less than or equal to: $X/(1 - U_{gp(i)}^{slow})$, assuming $U_{gp(i)}^{slow} < 1$. We now assume that the recurrence is true at rank p and show that it is true at rank $p+1$. Then, $W_i^{q^{(p+1)}}(t)$ meets:

$$\sum_{j \in gp(i)} \left(1 + \left\lfloor \frac{\max(0\,; W_i^{q^{(p)}}(t) - M_j^{1,q}) + J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + \sum_{j \in sp(i) \cup \{i\}} \left(1 + \left\lfloor \frac{t+J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + \mathcal{A}_i$$

$$\leq W_i^{q^{(p)}}(t) \cdot \sum_{j \in gp(i)} \frac{C_j^{slow}}{T_j} + \sum_{j \in gp(i)} \left(1 + \frac{J_j^1}{T_j}\right) \cdot C_j^{slow} + \sum_{j \in sp(i) \cup \{i\}} \left(1 + \left\lfloor \frac{t+J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + \mathcal{A}_i$$

$$\leq W_i^{q^{(p)}}(t) \cdot U_{gp(i)}^{slow} + X \quad \leq \quad X \cdot U_{gp(i)}^{slow}/(1 - U_{gp(i)}^{slow}) + X \quad = \quad X/(1 - U_{gp(i)}^{slow}).$$

The series $\mathcal{W}_i^q$ is non-decreasing and upper bounded. Hence, this series is convergent. ■

Lemma 8 *Let $\mathcal{S}$ be the ordered set of times $t = k \cdot T_j - J_j^1$, where $k \in \mathbb{N}$ and $j \in sp(i) \cup \{i\}$. Let t_1 and t_2 be two consecutive times of $\mathcal{S}$. Then, $\forall\, t' \in [t_1, t_2)$, $W_i^q(t') = W_i^q(t_1)$.*

Proof: We consider the series $\mathcal{W}_i^q$ and prove the lemma by recurrence. Let $\mathcal{S}$ be the ordered set of times t such that: $t = k \cdot T_j - J_j^1$, where $k \in \mathbb{N}$ and $j \in sp(i) \cup \{i\}$. By definition, if t_1 and t_2 are two consecutive times of set $\mathcal{S}$: $\forall\, j \in sp(i) \cup \{i\}$, $\forall\, t' \in [t_1, t_2)$, $\lfloor (t' + J_j^1)/T_j \rfloor = \lfloor (t_1 + J_j^1)/T_j \rfloor$. Hence, the lemma is met at rank 0. Assuming that the recurrence is true at rank p, we show that it is true at rank $p+1$. Indeed, $W_i^{q^{(p+1)}}(t')$ equals:

$$\sum_{j \in gp(i)} \left(1 + \left\lfloor \frac{\max(0\,;\, W_i^{q^{(p)}}(t') - M_j^{1,q}) + J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + \sum_{j \in sp(i) \cup \{i\}} \left(1 + \left\lfloor \frac{t' + J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + \mathcal{A}_i$$

$$= \sum_{j \in gp(i)} \left(1 + \left\lfloor \frac{\max(0\,;\, W_i^{q^{(p)}}(t_1) - M_j^{1,q}) + J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + \sum_{j \in sp(i) \cup \{i\}} \left(1 + \left\lfloor \frac{t_1 + J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + \mathcal{A}_i$$

$$= W_i^{q^{(p+1)}}(t_1). \qquad \blacksquare$$

Lemma 9 *For any time $t \geq 0$, for any flow $\tau_i \in \tau$, we have: $W_i(t + B_i^{slow}) \leq W_i(t) + B_i^{slow}$, where B_i^{slow} denotes the length of the longest busy period of level P_i on node slow.*

Proof: We consider the series $\mathcal{W}_i^q$ and prove the lemma by recurrence. By definition, we have: $B_i^{slow} = \sum_{j \in gp(i) \cup sp(i) \cup \{i\}} \lceil B_i^{slow}/T_j \rceil \cdot C_j^{slow}$. Moreover, as $\forall\, (a, b)$, $\lfloor a+b \rfloor \leq \lfloor a \rfloor + \lceil b \rceil$, we get:

$$\sum_{j \in sp(i) \cup \{i\}} \left(1 + \left\lfloor \frac{t + B_i^{slow} + J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} \leq \sum_{j \in sp(i) \cup \{i\}} \left(1 + \left\lfloor \frac{t + J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + \sum_{j \in sp(i) \cup \{i\}} \left\lceil \frac{B_i^{slow}}{T_j} \right\rceil \cdot C_j^{slow}$$

$$\leq \sum_{j \in sp(i) \cup \{i\}} \left(1 + \left\lfloor \frac{t + J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + B_i^{slow}.$$

Hence, the lemma is met at rank 0: $W_i^{q^{(0)}}(t + B_i^{slow}) \leq W_i^{q^{(0)}}(t) + B_i^{slow}$. Assuming that the recurrence is true at rank p, we get $W_i^{q^{(p+1)}}(t + B_i^{slow})$ equals to:

$$\sum_{j \in gp(i)} \left(1 + \left\lfloor \frac{\max(0\,;\, W_i^{q^{(p)}}(t + B_i^{slow}) - M_j^{1,q}) + J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + \sum_{j \in sp(i) \cup \{i\}} \left(1 + \left\lfloor \frac{t + B_i^{slow} + J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + \mathcal{A}_i$$

$$\leq \sum_{j \in gp(i)} \left(1 + \left\lfloor \frac{\max(0\,; W_i^{q(p)}(t) - M_j^{1,q}) + B_i^{slow} + J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + \sum_{j \in sp(i) \cup \{i\}} \left(1 + \left\lfloor \frac{t + B_i^{slow} + J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + \mathcal{A}_i$$

$$\leq \sum_{j \in gp(i)} \left(1 + \left\lfloor \frac{\max(0\,; W_i^{q(p)}(t) - M_j^{1,q}) + J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + \sum_{j \in sp(i) \cup \{i\}} \left(1 + \left\lfloor \frac{t + J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + B_i^{slow}$$

$$\leq W_i^{q(p+1)}(t) + B_i^{slow}. \quad \blacksquare$$

5.4 Worst case end-to-end response time

The worst case end-to-end response time of any packet m belonging to flow τ_i is equal to the latest starting time of m in node q, plus C_i^q, minus t, the arrival time of packet m in the distributed system. More precisely, it is equal to: $W_i^q(t) + C_i^q - t$. The worst case end-to-end response time of flow τ_i is equal to the maximum of the worst case end-to-end response times of its packets. Moreover, to compute this worst case response time, we have only to consider times t equal to: $k \cdot T_j - J_j^1 < B_i^{slow}$, with $k \in \mathbb{N}$ and $j \in sp(i) \cup \{i\}$ (see Lemmas 8 and 9).

Property 4 *When all the flows follow the same line $\mathcal{L}$ consisting of q nodes numbered from 1 to q and are scheduled according to the Fixed Priority algorithm, flows having the same priority being scheduled FIFO, the worst case end-to-end response time of any flow τ_i meets:*

$R_{max}{}_i^{\mathcal{L}} = \max_{t \in \mathcal{S}'} \{W_i^q(t) + C_i^q - t\}$, *where:*

$$W_i^q(t) = \sum_{j \in hp(i)} \left(1 + \left\lfloor \frac{\max\left(0; W_i^q(t) - M_j^{1,q}\right) + J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + \sum_{j \in sp(i) \cup \{i\}} \left(1 + \left\lfloor \frac{t + J_j^1}{T_j} \right\rfloor\right) \cdot C_j^{slow} + \mathcal{A}_i,$$

$$\mathcal{A}_i = \sum_{\substack{h=1 \\ h \neq slow}}^{q} C_{max,i}^h - C_i^q + H_i^{1,q} + (q-1) \cdot L_{max};$$

$\mathcal{S}'$ denotes the ordered set of times t such that: $-J_i^1 \leq t = k \cdot T_j - J_j^1 < B_i^{slow}$, with $k \in \mathbb{N}$ and $j \in sp(i) \cup \{i\}$.

Proof: This property can be deduced from Lemmas 7, 8 and 9. $\blacksquare$

It is important to notice that the cardinal of set S' only depends on flows having the same priority as τ_i and flow τ_i. We can also notice that in the single node case, this bound is exact.

Lemma 10 *In the single node case, the bound given by property 4 is this given in Property 2. Indeed, we get:*

$$W_i^1(t) = \sum_{j \in gp(i)} \left(1 + \left\lfloor \frac{W_i^1(t)+J_j^1}{T_j} \right\rfloor\right) \cdot C_j^1 + \sum_{j \in sp(i)} \left(1 + \left\lfloor \frac{t+J_j^1}{T_j} \right\rfloor\right) \cdot C_j^1 + \left\lfloor \frac{t+J_i^1}{T_i} \right\rfloor \cdot C_i^1 + \max(0\,;\, C^1_{\overline{max},i} - 1).$$

5.5 Evaluation of the delay due to packets belonging to $\overline{hp}(i)$

We improve in this section the bound given in Lemma 6 on the maximum delay incurred by τ_i directly due to packets belonging to $\overline{hp}(i)$ in the particular case when network delay is constant and the processing time on node h is equal to C^h for any flow. This case is frequently encountered as along a path, all the packets have the same size, equal to the minimum *Maximum Transmission Unit* (MTU) on the path, leading to avoid the segmentation and reassembling on the intermediate nodes. This alleviates the processing on any intermediate node.

For instance, in IPv6, the Path MTU Discovery determines this minimum MTU. In the case of a single line, this leads to consider a processing time equal to C^h for any flow on any node h. Moreover, point-to-point links are an example of links providing constant network delays.

Lemma 11 *On any node $h \in [1,q]$, the processing time of any flow being equal to C^h and the network delay being constant, the interarrival time of two successive packets on any node $h \in (1,q]$ is at least equal to* $\max_{k=1..h-1}\{C^k\}$.

Proof: As the network delay is constant, the interarrival time of two successive packets on any node $h \in (1,q]$ is equal to their interdeparture time on node $h-1$. Let k be the slowest node among nodes 1 to $h-1$. As the processing delay on any node is constant, the interdeparture time on node k is at least equal to C^k. As on any node $k+1$ to $h-1$, the processing time is less than or equal to C^k, the interarrival time on node h is at least equal to C^k. ■

Lemma 12 *When, on any node $h \in [1,q]$, the processing time of all flows is equal to C^h and the network delay is constant, then for any flow τ_i visiting nodes 1 to $h+1$, $h \in [1,q)$, the maximum delay directly due to packets belonging to $\overline{hp}(i)$ meets:*

$$\begin{cases} H_i^{1,h+1} = H_i^{1,h} \quad if \quad C^{h+1} \leq \max_{k=1..h}\{C^k\} \\ H_i^{1,h+1} \leq H_i^{1,h} + \max\left(0; C^{h+1}_{\overline{max},i} - 1\right) \quad otherwise, \end{cases}$$

where $H_i^{1,1} = \max\left(0; C^1_{\overline{max},i} - 1\right)$.

Proof: We assume that on any node $h \in [1,q]$, the processing time of all flows is equal to C^h and the network delay is constant. We consider a flow τ_i and distinguish two cases:

- $C^{h+1} \leq \max_{k=1..h} C^k$. By applying Lemma 11, no additional delay due to packets belonging to $\overline{hp}(i)$ is incurred by τ_i. Hence, $H_i^{1,h+1} = H_i^{1,h}$.
- $C^{h+1} > \max_{k=1..h} C^k$. By applying Lemma 11, an additional delay due to packets belonging to $\overline{hp}(i)$ is incurred by τ_i on node $h+1$. Hence, we have: $H_i^{1,h+1} \leq H_i^{1,h} + \max\left(0; C_{\overline{max},i}^{h+1} - 1\right)$. ■

6 Comparative evaluation in the distributed case

We first recall the computation principle of the worst case end-to-end response time in the distributed case when applying the holistic approach. Then, we give several examples that illustrate how close our results are to the exact results. We also compare our results to these obtained by the holistic approach.

6.1 Worst case end-to-end response time by the holistic approach

We now apply the holistic approach to compute the worst case end-to-end response time of any flow τ_i, when all flows follow the same line $\mathcal{L}$. We denote $Rmax_j^h$ (respectively $Rmin_j^h$) the maximum (respectively the minimum) response time experienced by packets of any flow τ_j in node h and J_j^h its worst case jitter when entering node h.

The holistic approach proceeds iteratively and starts with node 1. Knowing the value of J_j^1 for any $j \in [1,n]$, we compute $Rmax_j^1$ using Property 2 and $Rmin_j^1 = C_j^1$, $\forall\, j \in [1,n]$. We proceed in the same way for any node h, $h \in (1,q]$.

Knowing the value of $J_j^h = \sum_{k=1..h-1}(Rmax_j^k - Rmin_j^k) + (h-1)\cdot(L_{max} - L_{min})$, $\forall j \in [1,n]$, we compute $Rmax_j^h$ using Property 2 and $Rmin_j^h = C_j^h$.

A bound on the end-to-end response time of flow τ_i is given by:

$$\sum_{h=1}^{q} Rmax_i^h - \sum_{h=2}^{q} J_i^h + (q-1)\cdot L_{max}.$$

6.2 Examples

In this section, we give examples of bounds on the end-to-end response times of flows in a distributed system, when all flows follow the same line consisting of 5 nodes. We assume that $\tau = \{\tau_1, \tau_2, \tau_3, \tau_4, \tau_5\}$, all the flows having a period equal to 36 and entering the distributed system without jitter. The load is equal to 83,33%. Moreover, there is only one flow in the highest priority 3, whereas two flows share priority 2 and priority 1. Finally, we have $L_{max} = L_{min} = 1$.

As part of this paper, a simulation tool has been developed for providing the exhaustive solution of a real-time scheduling problem in a network. Indeed, once the different parameters have been specified, all the possible scenarios are generated and traffic feasibility is checked for each of them. The simulation result is a file containing the exact worst case end-to-end response time of each flow.

We consider four configurations: the processing times of the packets (i) decrease node after node, (ii) increase node after node, (iii) are not ordered and (iv) are the same in any node visited. The following tables give for each configuration and for any flow τ_i, $i \in [1,5]$, the exact value of its worst case end-to-end response time and the value computed according to the trajectory approach. To show the improvement of our results compared with those obtained by the classical technique, we also include in these tables the value computed according to the holistic approach.

Table 3. Worst case end-to-end response time of any flow τ_i, $i \in [1,5]$

(i) Processing times decrease node after node
$C_i^1 = 6$, $C_i^2 = 5$, $C_i^3 = 4$, $C_i^4 = 3$, $C_i^5 = 2$

Flow	P_i	Exact	Trajectory	Holistic
τ_1	1	48	48	200
τ_2	1	48	48	200
τ_3	2	41	41	86
τ_4	2	41	41	86
τ_5	3	29	29	39

(ii) Processing times increase node after node
$C_i^1 = 2$, $C_i^2 = 3$, $C_i^3 = 4$, $C_i^4 = 5$, $C_i^5 = 6$

Flow	P_i	Exact	Trajectory	Holistic
τ_1	1	48	48	261
τ_2	1	48	48	261
τ_3	2	45	51	85
τ_4	2	45	51	85
τ_5	3	36	39	39

(iii) Processing times are not ordered
$C_i^1 = 3$, $C_i^2 = 5$, $C_i^3 = 2$, $C_i^4 = 6$, $C_i^5 = 4$

Flow	P_i	Exact	Trajectory	Holistic
τ_1	1	48	48	238
τ_2	1	48	48	238
τ_3	2	44	47	89
τ_4	2	44	47	89
τ_5	3	34	35	39

(iv) Processing times are the same in any node
$C_i^1 = 6$, $C_i^2 = 6$, $C_i^3 = 6$, $C_i^4 = 6$, $C_i^5 = 6$

Flow	P_i	Exact	Trajectory	Holistic
τ_1	1	58	58	622
τ_2	1	58	58	622
τ_3	2	51	51	149
τ_4	2	51	51	149
τ_5	3	39	39	59

We observe that the values provided by the trajectory approach are exact for all flows in configurations (i) and (iv). In configurations (ii) and (iii), the values provided by the trajectory approach are exact only for the lowest priority flows. This can be explained by the overestimation of $H_i^{1,q}$, the delay directly due to flows belonging to $\overline{hp}(i)$, in such configurations.

The bounds provided by the holistic approach are very pessimistic for small priority flows. For instance, in configuration (iv), the value provided by the holistic approach is 11 times the exact one. Even for the highest priority flow, the bound can be pessimistic, as we can see in (iv), where it is equal to 1.5 time the exact value.

7 Conclusion

In this paper, we have established new results for Fixed Priority scheduling in the uniprocessor and distributed cases. By assuming that flows sharing the same priority are scheduled FIFO, we have revisited classical results in a uniprocessor context. As our solution enables to improve the worst case response times, any set of flows feasible with the classical solution is feasible with ours. The converse is false, as shown by an example. Moreover, we have determined the conditions leading to shorter response times. Then, we have established new results in a distributed context. We have shown how to compute an upper bound on the end-to-end response time of any flow with a worst case analysis using the trajectory approach. Thus, we have determined the worst case end-to-end response time of any flow and we have compared these results with the exact values and those provided by the classical holistic approach. We have shown that the bound given by the trajectory approach is reached in various configurations, whereas the holistic approach provides only a bound that can be very pessimistic.

References

[1] K. Tindell, A. Burns, A. J. Wellings, *Analysis of hard real-time communications*, Real-Time Systems, Vol. 9, pp. 147-171, 1995.

[2] J. Liu, *Real-time systems*, Prentice Hall, New Jersey, 2000.

[3] K. Jeffay, D. F. Stanat, C. U. Martel, *On non-preemtive scheduling of periodic and sporadic tasks*, IEEE Real-Time Systems symposium, pp. 129-139, San Antonio, USA, December 1991.

[4] L. George, N. Rivierre, M. Spuri, *Preemptive and non-preemptive scheduling real-time uniprocessor scheduling*, INRIA Research Report No 2966, September 1996.

[5] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, *An architecture for Differentiated Services*, RFC 2475, December 1998.

[6] R. Braden, D. Clark, S. Shenker, *Integrated services in the Internet architecture: an overview*, RFC 1633, June 1994.

[7] S. Baruah, R. Howell, L. Rosier, *Algorithms and complexity concerning the preemptive scheduling of periodic real-time tasks on one processor*, Real-Time Systems, 2, p 301-324, 1990.

[8] V. Sivaraman, F. Chiussi, M. Gerla, *End-to-end statistical delay service under GPS and EDF scheduling: a comparaison study*, INFOCOM'2001, Anchorage, April 2001.

[9] M. Vojnovic, J. Le Boudec, *Stochastic analysis of some expedited forwarding networks*, INFOCOM'2002, New York, June 2002.

[10] F. Chiussi, V. Sivaraman, *Achieving high utilization in guaranteed services networks using early-deadline-first scheduling*, IWQoS'98, Napo, California, May 1998.

[11] L. George, D. Marinca, P. Minet, *A solution for a deterministic QoS in multimedia systems*, International Journal on Computer and Information Science, Vol.1, N3, September 2001.

[12] K. Tindell, J. Clark, *Holistic schedulability analysis for distributed hard real-time systems*, Microprocessors and Microprogramming, Euromicro Journal, Vol. 40, 1994.

[13] J. Le Boudec, P. Thiran, *Network calculus: A theory of deterministic queuing systems for the Internet*, Springer Verlag, LNCS 2050, September 2003.

[14] A. Parekh, R. Gallager, *A generalized processor sharing approach to flow control in integrated services networks: the multiple node case*, IEEE ACM Transactions on Networking, Vol.2, N2, 1994.

[15] L. Georgiadis, R. Guérin, V. Peris, K. Sivarajan, *Efficient network QoS provisioning based on per node traffic shaping*, IEEE/ACM Transactions on Networking, Vol. 4, No. 4, August 1996.

[16] V. Sivaraman, F. Chiussi, M. Gerla, *Traffic shaping for end-to-end delay guarantees with EDF scheduling*, IWQoS'2000, Pittsburgh, June 2000.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399